

## **Rapport 4eme séance :** **10/01/2022**

Projet : Med Dispenser

### **Avant la séance**

J'ai réalisé la maquette en carton de la petite plaque que l'on fixe sur le moteur pour pousser les médicaments.

### **1ere partie de la séance**

Dans notre montage de l'horloge, écran et moteur nous avons ajouté la Led mais cela n'a pas fonctionné, en effet bien que celle-ci s'allumait, l'écran n'affichait plus les bonnes choses et le moteur ne tournait plus. Nous avons décidé de s'occuper de la Led plus tard.

Je me suis donc occupée du programme du moteur, c'est-à-dire le faire fonctionner à une heure précise. Je suis parvenue à faire ce code mais j'ai rencontré un problème : le moteur se met en route 1min avant l'heure indiquée.

J'ai rencontré un autre problème : j'ai voulu essayer de faire en sorte que le moteur fasse un seul tour à l'heure voulue, pour cela j'ai réduit sa vitesse grâce à la ligne de code :

monMoteur.setSpeed(1);

Mais cela n'a pas marché : avec cette ligne le moteur devrait avoir une vitesse qui lui permet de faire un tour par minute, mais il en fait plusieurs jusqu'à ce que l'heure change, à ce moment-là il s'arrête.

### **Programmes**

#### **Code principal**



```
codeparent horlogeecran moteur
//code principal
#include <RTClib.h>
RTC_DS1307 RTC;

void setup() {
  // put your setup code here, to run once:
  initHour();
  initMotor();
}

void loop() {
  // put your main code here, to run repeatedly:
  printHour();
  delay(3000);
  DateTime now=RTC.now();
  if ((now.hour()==17) && (now.minute()==21)){
    runMotor();}
  //runMotor();
}
```

## Code moteur

```
codeparent  horlogeecran  moteur
#include <Stepper.h>
#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11
int temps=10;
int i=0;
const int nombrePas=32*64;
Stepper monMoteur(nombrePas,8,9,10,11);

void initMotor() {
  Serial.begin(115200);
  monMoteur.setSpeed(1);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}

void runMotor() {
  for (i=0; i<=511; i++){
    Serial.println(i);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(temps);

    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(temps);

    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    delay(temps);

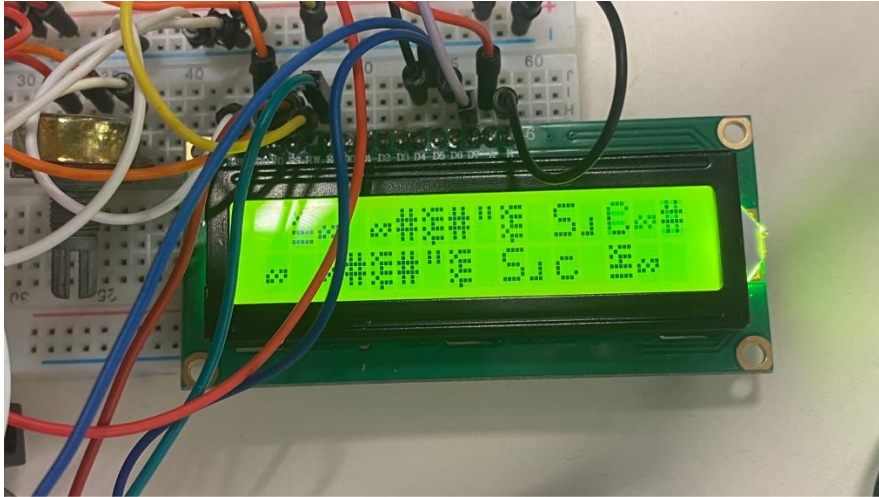
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    delay(temps);
  }
}
```

## 2eme partie de la séance

Avant notre passage à l'oral nous avons fait des simulations avec notre maquette, le moteur et des pilules. Nous avons donc vérifié que le système pour faire tomber les pilules fonctionnait correctement.

Nous avons aussi remarqué qu'au bout d'un certain temps l'affichage change.

### Changement de l'affichage

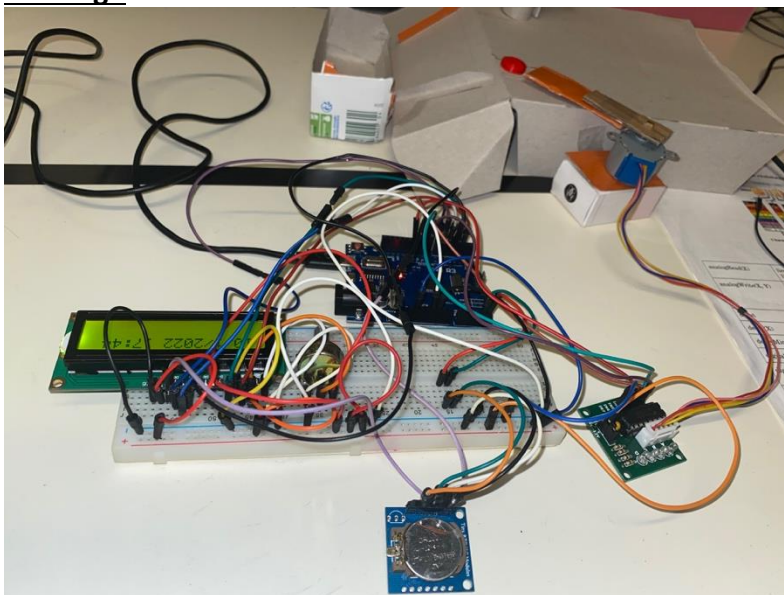


Après notre passage à l'oral, nous avons récupéré un afficheur I2C LCD 16x2 qui nous permettra d'enlever quelques fils. Nous allons aussi remplacer nos deux moteurs pas à pas par deux servomoteurs, ce qui nous fera utiliser 4 fils au lieu de 8. Durant le temps qu'il restait, j'ai donc cherché sur internet comment fonctionnent l'afficheur et les servomoteurs, j'ai ainsi pu trouver quelques exemples de codes et de montages.

(quelques sites que j'ai trouvé : <https://www.gotronic.fr/pj2-sbc-lcd16x2-fr-1441.pdf>, <https://arduino-france.site/lcd-1602/>, <https://arduino.developpez.com/tutoriels/arduino-a-l-ecole/?page=projet-12-utiliser-un-servomoteur>)

J'ai également installé la bibliothèque *LiquidCrystal\_I2C* pour pouvoir utiliser l'afficheur.

### Montage



Durant la prochaine séance, il faudra connecter les servomoteurs ainsi que l'afficheur, modifier les programmes pour les faire fonctionner avec ces nouveaux éléments et comprendre pourquoi l'écran n'affiche plus l'heure et la date au bout d'un certain temps.